

RECEIVED
CENTRAL FAX CENTER

SEP 27 2006

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Transmittal for Appeal Brief	Application Number: 10/600221 Filing Date: 06-19-03 First Named Inventor Hrle Group Art Unit 2162 Examiner G. Colan Atty. Docket Num. SVL920030011US1
-------------------------------------	--

Commissioner for Patents
By Fax to 571-273-8300

A fee of \$500.00 is due with this response. The Commissioner is hereby authorized to charge payment of the following fees associated with this communication or credit any overpayment to **Deposit Account 09-0460**.

Attached hereto is a Brief (totaling 25 pages) in support of applicants' appeal to the Board of Patent Appeals and Interferences from the final rejection of applicants' claims by the Examiner in an Office Action dated May 19, 2006.



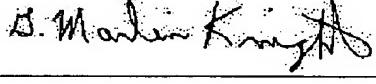
G. Marlin Knight (#33,409)
Attorney of Record
PO Box 1320
Pioneer, CA 95666
209-295-1982

Transmission by Fax CERTIFICATE

I hereby certify that the above paper is being transmitted to telephone no. 571-273-8300 on the date indicated below and is addressed to the Commissioner for Patents.

Date of Transmission: September 27, 2006

Person sending paper: G. Marlin Knight

Signature: 

10/600221

1

SVL920030011US1

RECEIVED
CENTRAL FAX CENTER

SEP 27 2006

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appeal Brief by Applicants	Application Number: 10/600221 Filing Date: 06-19-03 First Named Inventor Hrle Group Art Unit 2162 Examiner G. Colan Atty. Docket Num. SVL920030011US1
-----------------------------------	---

Commissioner for Patents

1. Real Party in Interest:

As the assignee of all rights in the patent application, the following designates the Real Party in Interest:

INTERNATIONAL BUSINESS MACHINES CORPORATION, a corporation of New York, having a place of business at Armonk, New York 10504.

10/600221

1

SVL920030011US1

2. Related Appeals and Interferences: None.

10/600221

2

SVL920030011US1

3. Status of Claims:

The claims in this appeal are claims 1-33. Each of these claims has been finally rejected.

10/600221

3

SVL920030011US1

4. Status of Amendments:

All amendments have been entered.

10/600221

4

SVL920030011US1

5. Summary of the Claimed Subject Matter

The claimed invention relates to methods and systems for performing backup of a database management system (DBMS) without suspending updates to the database by application programs. A backup according to the method of the invention can be used to restore the DBMS to the time of the backup or for a system level point-in-time (PIT) recovery for backing out application program's errors using the live system's logs. Specification p. 3, lines 1-7; Figure 1. The independent claims 1, 9 and 17 cover a DBMS, and claim 26 covers computer usable media with a computer program that implements the invention. The DBMS uses a write-ahead logging protocol and restores consistency between the log records and the data during a restart. The data and log records are stored on different storage volumes. Specification p. 3, lines 14-15, Figure 1. While a backup system lock is held by a backup utility, the DBMS continues updating objects in the normal fashion and only a limited number of actions are suspended. The actions that are suspended are those that change an external file system catalog, require writing updates that extend across a storage volume boundary; and updating the REDO log point. Specification p. 3, lines 20-26; Figure 2. The REDO log point in checkpoint information is frozen while the backup system lock is taken by the backup utility. Specification p. 3, lines 25-26; Figure 2.

6. Grounds of Rejection

Claims 1-33 were rejected under section 103(a) as being unpatentable over Kawamura, et al. 5778388, in view of Mosher, et al. 6785696.

10/600221

6

SVL920030011US1

7. Arguments

The Examiner rejected claims 1-33 under section 103(a) as being unpatentable over Kawamura, et al. 5778388, in view of Mosher, et al. 6785696. The Examiner cited Kawamura for all of the elements of the independent claims except for including a REDO point in checkpoint information for which Mosher was cited (col. 8, lines 66-67 and 1-3). Applicants respectfully disagree.

Each of applicants' independent claims 1, 9, 17, and 26 involve the use of a backup system lock and describe system actions while the backup system lock is taken. Claim 1 explicitly recites that the backup system lock is taken by a backup utility. It is respectfully submitted that neither of the cited reference teaches the specific actions claimed. The word "backup" does not even appear in searchable text of the Kawamura patent, so it is clear that Kawamura is describing the standard operation of the database program, not a backup process. Therefore, Kawamura is addressing a completely different subject matter than is addressed by the applicants' claims.

Applicants submit that the only overlap between Kawamura and the present claims are general references to the use of locks in DBMS processing. A lock is a general software tool used to serialize multi-tasking, so locks have an unlimited number of uses. The use of locks during the normal operation of the DBMS and during backup operations is well known. The applicants' claims are not broad claims on the use of locks in general. The applicants claim a very specific set of actions by the DBMS when a backup system lock is taken.

Applicants' claim 1 makes it clear that the backup system lock is held by a backup utility. Claim 1 includes the following language:

while a backup system lock is held by a backup utility, continues updating objects except for suspending actions that change an external file system catalog, suspending writing updates of objects that extend across a storage volume boundary; and freezing a REDO log point in checkpoint information while the backup system lock is taken by the backup utility.

Thus, the claim describes specific actions taken by the DBMS while the backup system

10/600221

7

SVL920030011US1

lock is taken. Applicants' invention allows more efficient execution of the backup process by prohibiting only certain actions while it is in progress. Except for these limited actions, the database operation is allowed to continue normally.

The Kawamura reference is not describing processing during a backup operation of any kind; therefore, the teachings of Kawamura are not on point. The Examiner specifically referenced Kawamura col. 9, lines 33-35 and col. 9, lines 52-56 for the elements of claim 1 cited above. The entire paragraph encompassing col. 9, lines 33-35 is:

Since the pages to be written in the database at the syncpoint are confirmed, the buffer pool is immediately unlocked (step 256). While the buffer pool is in the locked state, any input and output operations are inhibited for the external storages. This consequently leads to a short lock period of time in which only the CPU processing is executed. Thanks to this provision, the processing of transactions awaiting the unlocking of the buffer pool can be continuously executed. According to the output page control table list thus produced, for the pages to be written in the database at the syncpoint, a write request is issued to the deferred write processing part 25, thereby initiating the operation to write the pages in the database (step 257).

The references are to figure 1 which makes it clear that the buffer pool lock has nothing to do with a backup process. Even if one ignores this critical difference, the Examiner's argument requires that Kawamura's buffer pool lock be to equivalent of applicants' "backup system lock" which is not the case. If the Examiner is equating Kawamura's buffer pool lock to applicants' backup system lock, the burden is the Examiner to show the equivalence, and the Examiner has failed to do so.

Moreover, applicants' claim 1 requires that the backup system lock is held by a backup utility, and this is also not taught by Kawamura. Applying Kawamura to applicants' claim requires 1) ignoring the context of his teaching, and 2) changing the meaning of his terms. Even after these invalid steps have been taken, the argument further requires ignoring the language in the claim of "updating objects except for" Even after all this, the argument fails to supply any support for "freezing a REDO log point in checkpoint information while the backup system lock is taken"

10/600221

8

SVL920030011US1

The Examiner has failed to cite any component in Kawamura's system that is a backup utility or its equivalent. Kawamura's buffer pool locks cannot be said to be the equivalent of backup system locks when Kawamura has no backup system. A similar mismatch between Kawamura's teaching and the applicants' claims appears when the actions associated with the respective locks are noted. The Examiner apparently interpreted Kawamura's teaching that input and output operations are inhibited for the external storage when the buffer pool is in the locked state as implicitly equivalent to applicants' claimed step of "continuing to update the data while the backup system lock is taken, except for suspending actions that change an external file system catalog, and except for suspending writing updates of objects that extend across a storage volume boundary." Again the burden is on the Examiner to show implicit or hidden equivalences, and the Examiner has failed to meet this burden.

The Examiner's interpretation is incorrect for several reasons including that Kawamura is not teaching a backup method and does not teach "continuing to update the data..." during a backup operation. When the "buffer pool is in the locked state" in Kawamura's system "any input and output operations are inhibited for the external storages." Thus, Kawamura's system does not continue "updating objects except for ..." the specific categories in the applicants' claims. Because Kawamura is locking out all writing to external storage, he has no teaching that distinguishes between actions that change an external file system catalog and updates of objects that extend across a storage volume boundary.

If we use the test of whether applicants' claims read on Kawamura, we immediately see that the Examiner's interpretation requires that we ignore the words "updating objects except for ..." in applicants' claim. There is nothing in Kawamura to which the branching or decision making process covered by the phrase "... except for ..." can be applied. As will be discussed below the addition of the Mosher reference does not cure this error in the Examiner's case.

While applicants' claims allows the system to continue, in most cases, to write to the external storage during the backup process, Kawamura prohibits all writing when the buffer lock is taken. Therefore, it is clear that the cited sections of Kawamura is describing normal transaction processing, not processing during a backup operation of
10/600221

any kind, and, moreover, they do not describe the specific processing that occurs when a backup system lock is taken as applicants claim.

It is respectfully submitted that the Examiner has erroneously interpreted the general buffer pool locks described in Kawamura as being locks used during backup process. For example, the Examiner cited to Kawamura col. 9, lines 7-9 as teaching a backup system lock. The cited section is:

When the lock is reserved for the buffer pool, accesses to the buffer pool due to transactions are temporarily maintained in the wait state until the locked state is released.

Buffer pool locks are not backup system locks and, moreover Kawamura's buffer pool lock is not taken by a backup program according to applicants' claim 1. This is not just a matter of the name applied to the lock because the actions claimed by the applicants while the lock is taken are different from the actions taught by Kawamura. Kawamura's buffer pool locks are simply used during the normal DBMS update procedure:

When the database management system conducts an exclusive control operation at the row level, a lock is obtained in the exclusive mode for the row specified as an object of the update operation (step 61). When the lock is reserved for the row, a page input request is issued to the buffer pool control part 23 in the exclusive mode for a page in which the row is stored (step 62). (col. 7, line 64 through col. 8, line 3.)

Therefore, Kawamura does not teach using the buffer pool locks to trigger the same set of actions that applicants claim.

The second reference Examiner relied on is the Mosher reference. Unlike Kawamura, Mosher does relate to a backup process including backing up primary nodes onto backup nodes. The Examiner cited Mosher col. 8, lines 66-67 and 1-3 for including a REDO point in checkpoint information. Therefore, the Examiner did not rely on Mosher to supply to missing teaching on processing during backup system locks. Mosher is non-analogous art to Kawamura since Mosher deals with a method and system for backing up primary nodes onto backup nodes where the primary nodes can

each originate a distributed transaction and can participate in a distributed transaction. (See Abstract). Mosher cannot be reasonably combined with Kawamura, since they are dealing with different subjects.

Moreover, even if the two references are combined, applicants' claimed invention does not result. Because Kawamura does not have a backup system lock, there is no obvious way for Mosher's system to interact with Kawamura's. The Examiner is suggesting that one of ordinary skill in the art would be motivated to combine Kawamura and Mosher by having the backup system takes Kawamura's buffer pool lock, but there is nothing in either reference that justifies this conclusion. It is only using applicants' specification in hindsight that such an action would ever come to mind and then only if the buffer pool locks are misinterpreted. Moreover, if a backup program did take a buffer pool lock in Kawamura's system, the result would still not be the claimed invention. In addition to points already noted, the system would still not include the freezing the REDO log point because this teaching is not found in either of the cited references.

Applicants' claim 1 includes "freezing a REDO log point in checkpoint information while the backup system lock is taken by the backup utility." The three other independent claims 9, 17 and 26 contain similar language. The Examiner cited Mosher col. 8, lines 43-46 and 50-52 for a general teaching on REDO log points. However, Mosher is not teaching freezing a REDO log point for any reason and, therefore, certainly is not teaching freezing the REDO when a backup system lock is taken. The claimed action is not to the general use of a REDO log point, but rather to specifically freezing the REDO log point while a backup lock is taken in the context of the other elements of the claim. It is respectfully submitted that it is only using the applicants' specification in hindsight that one would find any relationship between the REDO log point and a backup system lock. The idea of freezing the REDO log point is found only in the applicants' specification, not in any of the cited references.

Thus, neither Mosher nor Kawamura teach the claimed use of a backup system lock nor the specific actions of suspending only actions that change an external file system catalog, and suspending writing updates of objects that extend across a storage volume boundary during the backup system lock. In addition, the freezing the REDO log point while a backup system lock is taken is not taught by Mosher or Kawamura.

10/600221

11

SVL920030011US1

Dependent claim 2 further distinguishes over the cited references by adding that the database management system further comprise "a backup utility that obtains the backup system lock before starting a backup process; copies the first set of storage volumes to a first set of backup volumes; records information identifying the first set of backup volumes in a dataset." The Examiner cites to Kawamura figure 14 item 408; col. 11, lines 9-10; figure 14 item 409; lines 10-12 (with no column referenced, but presumably meant to be col. 11); and col. 6, lines 34-38. As already noted Kawamura does not teach a backup process at all, so this reference is unjustified. Similarly item 408 in figure 14 is a block that says nothing more than "Unlock Buffer Pool." Similarly item 409 in figure 14 is a block that says nothing more than "Write Page on Disk." The referenced sections are devoid of any action that corresponds to applicants' claim element: "copies the first set of storage volumes to a first set of backup volumes." The referenced sections are also devoid of any action that corresponds to applicants' claim element: "records information identifying the first set of backup volumes in a dataset." The referenced section at col. 6, lines 34-38 refers to log sequence number (LSN) fields that have no application to the subject matter of the claim.

Dependent claim 3 further distinguishes over the cited references by adding that the "backup utility copies the second set of storage volumes to a backup medium; and records backup volume information for the second set of storage volumes in the dataset." Kawamura includes no teaching about backup utilities and, therefore, does not teach having backup utility copy the second set of storage volumes to a backup medium. Similarly, there is no teaching in Kawamura of "backup volume information for the second set of storage volumes;" therefore, Kawamura also does not teach copying such information as claimed. The Examiner cites Kawamura's figures 2 and 21, but these figures do not include any reference to backup utilities. Figure 21, item 2710 cited by the Examiner against claim 3, only says "Write Log Data in Log File."

The Examiner cited to Mosher col. 4, lines 26-27):

The MIT 70, 72 generally contains timing and transaction state audit records while the SIT 74a-b, 76a-b generally contains the update and undo audit records. The Purger Process periodically deletes image trail files that are not needed by the backup system. col. 4, lines 25-29.

Thus, the addition of Mosher fails to provide the teachings that are lacking in Kawamura.

Dependent claim 4 further distinguishes over the cited references by adding that a restore utility performs a point-in-time recovery using the data from the first set of backup volumes, a user specified point-in-time, and the logs on the second set of storage volumes. The Examiner cites Mosher figure 8, item 366 "RTD Timestamp" which fails to provide the missing elements from the previous claims.

Dependent claim 5 further distinguishes over the cited references by adding that "the restore utility marks a first object as recovery-pending when a log record identifies the first object as having been updated without log records so that subsequent restoration of the first object can be made from an image copy." The Examiner cited Mosher col. 10, lines 26-28 and 56-61 that is part of the section titled Creating the Local Undo List (Detail B):

Next, in step 530, the SIT is traversed backwards from its EOF until the EndMAT position is reached, during which traversal, an entry, marked as "unknown," is added to the TST if there was no previous transaction information in the TST for the record in the SIT, and the synchronization information is updated, if the SIT being scanned is for a volume on which the synchronization file is stored. ... In step 550, a data record from the SIT is obtained and if the record is not for the volume with the synchronization file, as determined in step 552, an entry marked as "unknown" is added to the TST in step 554. The EndMAT position is tested to determine whether it has been reached or exceeded. If not the next record from the SIT is obtained, in step 550.

The cited section fails to teach marking an object as "recovery-pending" and fails to teach the use of log records identifies the first object as having been updated without 10/600221

log records and fails to teach restoration of the "recovery-pending" object from an image copy. Again it is only using the applicants' specification that any of these concepts could be stated because they do not appear in the references.

Dependent claim 6 further distinguishes over the cited references by adding that the mainline database system "writes log records to identify objects that have been updated without log records and writes log records to identify objects that have been created, extended and/or deleted." The Examiner cited Mosher col. 10, lines 24-29 for "log records to identify objects that have been updated without log records," but this section of Mosher is quoted above is not teaching anything about logging. In fact, the cited sections are dealing with completely different subjects with no overlap with applicants' claims. If there is an implicit relationship between Mosher's Transaction Status Table (TST) created the Purger and the log records recited in applicants' claim, the Examiner has failed to establish it.

Dependent claim 8 further distinguishes over the cited references by adding that the mainline database system "obtains the backup system lock before updating objects that change an external file system catalog or that extend across a storage volume boundary." The Examiner specifically referenced Kawamura col. 9, lines 33-34, which is quoted above. Since the referenced section only refers to buffer pool locks, this again reflects the mistaken argument that applicants' backup system lock is equivalent to Kawamura's buffer pool lock. In addition, Kawamura's teaching on when the DBMS would obtain the buffer pool lock does not match applicants' claimed conditions of updating objects 1) that change an external file system catalog or 2) that extend across a storage volume boundary.

The other dependent claims for independent claims are generally parallel to those discussed above, and Examiner's rejections of these claims are believed to be overcome by the arguments above.

Conclusion

Applicants respectfully submit that the foregoing arguments have shown that the Kawamura and Mosher references cited in support of the rejections do not teach applicant's claimed invention even if combined as the Examiner suggests. Applicants' claims are directed at a specific method of operating a backup of a database using a particular storage arrangement and allowing particular actions to continue during the backup. Applicants respectfully submit that the references singly and when combined fail to teach claimed elements of applicants' claims. Applicant further submits that the motivation to combine the selected features of the references is not present because they are dealing with different subjects.

Neither Mosher nor Kawamura teach the claimed use of a backup system lock to trigger the specific actions of suspending actions that change an external file system catalog and suspending writing updates of objects that extend across a storage volume boundary during the backup system lock. In addition, the freezing the REDO log point while a backup system lock is taken is not taught by Mosher or Kawamura.

The applicant, therefore, respectfully requests that the rejections be withdrawn and that the claims be allowed.

Respectfully Submitted,



G. Marlin Knight (#33,409)
PO Box 1320
Pioneer, CA 95666
209-295-1982

8. Appendix of Claims in the Appeal:**1. A database management system comprising:**

a mainline database system that makes modifications to data in the database management system using a write-ahead logging protocol; stores data on a first set of storage volumes and stores log records on a second set of storage volumes; restores consistency between the log records and the data during a restart, and while a backup system lock is held by a backup utility, continues updating objects except for suspending actions that change an external file system catalog, suspending writing updates of objects that extend across a storage volume boundary; and freezing a REDO log point in checkpoint information while the backup system lock is taken by the backup utility.

2. The database management system of claim 1 further comprising a backup utility that obtains the backup system lock before starting a backup process; copies the first set of storage volumes to a first set of backup volumes; records information identifying the first set of backup volumes in a dataset.**3. The database management system of claim 2, wherein the backup utility copies the second set of storage volumes to a backup medium; and records backup volume information for the second set of storage volumes in the dataset.****4. The database management system of claim 2 further comprising a restore utility that performs a point-in-time recovery using the data from the first set of backup volumes, a user specified point-in-time, and the logs on the second set of storage volumes.**

5. The database management system of claim 4 wherein the restore utility marks a first object as recovery-pending when a log record identifies the first object as having been updated without log records so that subsequent restoration of the first object can be made from an image copy.
6. The database management system of claim 1 wherein the mainline database system writes log records to identify objects that have been updated without log records and writes log records to identify objects that have been created, extended and/or deleted.
7. The database management system of claim 1 wherein the mainline database system stores checkpoint information periodically and the backup utility records a log apply starting point corresponding to a last checkpoint information storage point.
8. The database management system of claim 1 wherein the mainline database system obtains the backup system lock before updating objects that change an external file system catalog or that extend across a storage volume boundary.

9. A database management system, that performs a backup without suspending updates and the backup can be restored using a write-ahead logging protocol restart, comprising:

means for modifying data in the database management system using a write-ahead logging protocol;

means for restoring consistency between log records and the data during a restart;

means for storing data on a first set of storage volumes and storing log records on a second set of storage volumes;

means for freezing a REDO log point in checkpoint information while a backup system lock is taken; and

means for continuing to update the data while the backup system lock is taken, except for suspending actions that change an external file system catalog, and except for suspending writing updates of objects that extend across a storage volume boundary.

10. The database management system of claim 9 further comprising:

means for obtaining the backup system lock before starting a backup process; for copying the first set of storage volumes to a first set of backup volumes; and for recording information identifying the first set of backup volumes in a recovery control dataset and in an external file system's control dataset.

11. The database management system of claim 10 further comprising means for copying the second set of storage volumes to a second set of backup volumes; and means for recording information identifying the second set of backup volumes in the recovery control dataset and in the external file system's control dataset.

12. The database management system of claim 10 further comprising means for restoring data from the first set of backup volumes to the first set of storage volumes and for performing a point-in-time recovery using a user specified point-in-time, and the logs on the second set of storage volumes.
13. The database management system of claim 12 wherein the means for restoring data further comprises means for marking a first object as recovery-pending when a log record identifies the first object as having been updated without log records so that subsequent restoration of the first object can be made from an image copy; processing log records identifying a second object which has been newly created by allocating space for the second object; processing log records identifying a third object which has been newly extended by allocating additional space for the third object; processing log records identifying a fourth object which has been deleted by freeing space for the fourth object; and means for setting a mode to indicate that the point-in-time recovery has completed.
14. The database management system of claim 9 further comprising means for writing log records to identify objects that have been updated without log records and writing log records to identify objects that have been created, extended or deleted.
15. The database management system of claim 10 further comprising means for storing checkpoint information periodically and the means for backing up the data further comprises means for recording a log apply starting point corresponding to a last checkpoint information storage point.
16. The database management system of claim 9 further comprising means for obtaining the backup system lock before updating objects that change an external file system catalog or that extend across a storage volume boundary.

17. A method of operating a database management system comprising the steps of:
 - modifying data in the database management system using a write-ahead logging protocol;
 - restoring consistency between log records and the data during a restart;
 - storing data on a first set of storage volumes and storing log records on a second set of storage volumes;
 - freezing a REDO log point in checkpoint information while a backup system lock is taken; and
 - continuing to update the data while the backup system lock is taken, except for suspending actions that change an external file system catalog, and except for suspending writing updates of objects that extend across a storage volume boundary.
18. The method of claim 17 further comprising the step of backing up the data and wherein the step of backing up the data further comprises obtaining the backup system lock and after obtaining the backup system lock, copying the first set of storage volumes to a first set of backup volumes.
19. The method of claim 18 further wherein the step of backing up the data further comprises the step of recording information identifying the first set of backup volumes in a recovery control dataset and in an external file system's control dataset.
20. The method of operating the database management system of claim 18 further comprising backing up log records, after backing up the data, by copying the second set of storage volumes to a second set of backup volumes.
21. The method of operating the database management system of claim 18 further comprising the steps of restoring data from the first set of backup volumes to the first set of storage volumes and performing a point-in-time recovery using a user specified point-in-time, and the logs on the second set of storage volumes.

22. The method of operating the database management system of claim 21 wherein the step of performing a point-in-time recovery further comprises the steps of marking a first object as recovery-pending when a log record identifies the first object as having been updated without log records so that subsequent restoration of the first object can be made from an image copy; processing log records identifying a second object which has been newly created by allocating space for the second object; processing log records identifying a third object which has been newly extended by allocating additional space for the third object; processing log records identifying a fourth object which has been deleted by freeing space for the fourth object; and setting a mode to indicate that the point-in-time recovery has completed.
23. The method of operating database management system of claim 17 further comprising writing log records to identify objects that have been updated without log records and writing log records to identify objects that have been created, extended or deleted.
24. The method of operating the database management system of claim 18 further comprising storing checkpoint information periodically and the step of backing up the data further comprises recording a log apply starting point corresponding to a last checkpoint information storage point.
25. The method of operating the database management system of claim 17 wherein the step of continuing to update the data while a backup system lock is taken further comprises obtaining the backup system lock before updating objects that change an external file system catalog or that extend across a storage volume boundary.

26. An article of manufacture comprising computer usable media including at least one computer program recorded therein that is capable of causing a computer system to perform a method of operating a database management system comprising the steps of:

modifying data in the database management system using a write-ahead logging protocol;

restoring consistency between log records and the data during a restart;

storing data on a first set of storage volumes and storing log records on a second set of storage volumes;

freezing a REDO log point in checkpoint information while a backup system lock is taken; and

continuing to update the data while the backup system lock is taken, except for suspending actions that change an external file system catalog, and except for suspending writing updates of objects that extend across a storage volume boundary.

27. The article of manufacture of claim 26 wherein the method further comprises the step of backing up the data and wherein the step of backing up the data further comprises obtaining the backup system lock and after obtaining the backup system lock, copying the first set of storage volumes to a first set of backup volumes.

28. The article of manufacture of claim 27 wherein the step of backing up the data further comprises the step of recording information identifying the first set of backup volumes in a control dataset.

29. The article of manufacture of claim 27 wherein the method further comprises backing up log records, after backing up the data, by copying the second set of storage volumes to a second set of backup volumes.

30. The article of manufacture of claim 29 wherein the method further comprises recording information identifying the second set of backup volumes in the recovery control dataset and in the external file system's control dataset.

31. The article of manufacture of claim 27 wherein the method further comprises the steps of restoring data from the first set of backup volumes to the first set of storage volumes and performing a point-in-time recovery using a user specified point-in-time, and the logs on the second set of storage volumes.
32. The article of manufacture of claim 31 wherein the step of performing a point-in-time recovery further comprises the steps of marking a first object as recovery-pending when a log record identifies the first object as having been updated without log records so that subsequent restoration of the first object can be made from an image copy; processing log records identifying a second object which has been newly created by allocating space for the second object; processing log records identifying a third object which has been newly extended by allocating additional space for the third object; processing log records identifying a fourth object which has been deleted by freeing space for the fourth object; and setting a mode to indicate that the point-in-time recovery has completed.
33. The article of manufacture of claim 26 wherein the step of continuing to update the data while a backup system lock is taken further comprises obtaining the backup system lock before updating objects that change an external file system catalog or that extend cross a storage volume boundary.

9. Evidence Appendix: None

10/600221

24

SVL920030011US1

10. Related Proceedings Appendix: None

10/600221

25

SVL920030011US1